

## **REMARKS**

The following remarks and amendments attend to all rejections presented in the pending June 10, 2005 office action. Claims 1-20 remain pending in this application, with claims 1, 8, 14 and 19 being independent.

Claims 1, 2, 8, 9, 14, 15, 19 and 20 are amended for clarification: the term 'set of visited nets' is amended to use 'set of visited incomplete trace objects' as described in at least paragraph [0028] of the specification. No new matter is added.

### **In the Specification**

Paragraph [0001] of the specification is modified to replace attorney docket numbers of related applications with application numbers. No new matter is added.

### **Claim Objections**

Claims 1, 8, 14 and 19 stand objected to because the phrase "set of visited nets" appears to be a list, but is not claimed as a list.

Respectfully, the Examiner appears to be confused between (a) the incomplete trace stack 115, which may be implemented as a list as described in at least paragraphs [0022] and [0026], and (b) the set of visited incomplete trace objects 113, as described in at least paragraphs [0028] and [0032]. As shown in step 217 of FIG. 2 and described in paragraph [0028], "the incomplete trace object 110 ... is inserted into a set of visited incomplete trace objects 113." Incomplete trace object 110 "represents a particular (unique) net in a hierarchical design, and ... maintain information about what other connections in this net need to be traced" See paragraph [0025] of the specification.

The term 'set of visited nets' is used in FIGs. 2 and 5 with regard incomplete trace objects, each of which is clearly composed of a specified net and an instance history. See at least paragraph [0028] of the specification. For clarification, claims 1, 2, 8, 9, 14, 15, 19 and 20 are thus amended to replace the term 'set of visited nets' by the term 'set of visited incomplete trace objects'. No new matter is added.

Moreover, contrary to the Examiner's assertion on page 2 of the present office action, we contend that the use of 'set' structure is appropriate in view of the

specification. InstantWeb's Online Computing Dictionary defines a set as (emphasis added):

A collection of objects, known as the elements of the set, specified in such a way that we can tell in principle whether or not a given object belongs to it. E.g. the set of all prime numbers, the set of zeros of the cosine function. For each set there is a predicate (or property) which is true for (possessed by) exactly those objects which are elements of the set. The predicate may be defined by the set or vice versa. *Order and repetition of elements within the set are irrelevant.*

Now compare a 'set' to a list. In a set, order and repetition of elements within the set are irrelevant, whereas a list may contain more than one identical element and the order may be significant. Thus the immediate claims appropriately use the term 'set'.

#### **Claim Rejections – 35 U.S.C. §102**

Claims 1, 2, 8-9, 14-15, 19 and 20 stand rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,842,888 to Roberts (hereinafter "Roberts"). Respectfully, we disagree. To anticipate a claim, Roberts must teach every element of the claim and "the identical invention must be shown in as complete detail as contained in the ... claim." *MPEP 2131* citing *Verdegaal Bros. V. Union Oil Co. of California*, 814 F.2d 628, 2 USPQ2d 1051 (Fed. Cir. 1987) and *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 9 USPQ2d 1913 (Fed. Cir. 1989). Roberts does not teach every element of claims 1, 2, 8-9, 14-15, 19 and 20.

By way of example, the immediate application provides, among other features, a method for iteratively traversing a hierarchical circuit design. A mechanism is disclosed "for iteratively traversing a path through nets in a hierarchical design, without redundantly returning or identifying nets encountered in the trace process." See paragraph [0021] of the specification. Thus, in this example, the immediate application traces a path through nets in a hierarchical circuit design such that a net is not returned more than once. A set of visited incomplete trace objects is used to store information of nets that have not been traced such that these nets may be traced after

completion of a current trace. The immediate application does not necessarily traverse the complete hierarchy in order to identify all nets connected to an initial net.

On the other hand, Roberts discloses traversing “the entire hierarchy first, building a one-dimensional list and then going back and netlisting every item in that list.” See Roberts col. 2, lines 12-16. For example, Roberts, as shown in the pseudocode of col. 2, lines 35-43, uses a recursive function ‘BuildList’ containing a ‘foreach’ loop that visits each unique cell type in the hierarchy. Of particular note is that Roberts operates at the ‘cell’ level of a hierarchical database, and not at the portinst/port level as in the immediate application. Roberts does not traverse the hierarchical design by following a path through nets. As shown in at least FIG. 2, Roberts is concerned only with cells and parents/children of cells. Roberts does not disclose or suggest tracing of nets through portinsts/ports and does not disclose connectivity between cells. See, for example, Roberts FIG. 11. Thus, the traversing of Roberts is clearly different from that of the immediate application.

In particular, amended claims 1, 2, 8, 9, 14, 15, 19 and 20 recite methods, systems and software products that iteratively traverse a hierarchical circuit design; these claims include at least certain of the following claim elements:

- a) an iterator function;
- b) an incomplete trace object;
- c) a processor for executing the iterator function and invoking the incomplete trace object;
- d) selecting an initial net and an instance history that uniquely defines the initial net within the circuit design;
- e) appending, to a list of nets to be processed, the initial net and the instance history;
- f) inserting, into a set of visited incomplete trace objects, the initial net and the instance history;
- g) visiting, in response to a first request from a user, each additional net connected to the initial net;
- h) returning, in response to a second request from the user, the initial net and each additional net;

- i) determining, for each one of the nets to be processed in the list of nets to be processed, the additional nets that are connected to the net to be processed;
- j) checking the set of visited incomplete trace objects to determine unvisited additional nets that have not been visited;
- k) appending, to the list of nets to be processed, each of the unvisited additional nets; and
- l) inserting, into the set of visited incomplete trace objects, each of the unvisited additional nets.

Roberts does not disclose an iterator function, or an incomplete trace object, or a processor for executing the iterator function and invoking the incomplete trace object, as in elements a), b) and c) above. As described in paragraphs [0023-0033] of the specification and shown in FIG. 2 of the drawings, the hierarchical net iterator function utilizes incomplete trace objects for circuit design traversal. Roberts thus cannot anticipate features of elements a), b), c) (present in claim 14).

As noted above, Roberts does not disclose or suggest selecting *an initial net* within a hierarchical circuit design, as in element d) (present in claims 1, 8, 14 and 19). Roberts is concerned only with cells. Element e) requires that the initial net and the instance history be appended to a list of nets to be processed. Again, since Roberts does not disclose selection of nets, Roberts cannot anticipate element e) (present in claims 1, 8, 14 and 19).

Roberts also does not disclose or suggest a set of visited incomplete trace objects into which the initial net and instance history are inserted, as in element f) (present in claims 1, 8, 14 and 19).

Roberts further does not process nets and therefore cannot anticipate element g) (present in claims 1, 8, 14 and 19).

Roberts also does not disclose or suggest returning the initial net and each additional net connected to the initial net, as in element h); it cannot therefore anticipate requirement h) (present in claims 1, 14 and 19).

Roberts does not determine additional nets that are connected to each one of the nets to be processed in the list of nets to be processed and cannot therefore anticipate element i) (present in claims 2, 9, 15 and 20).

Roberts does not disclose a set of visited incomplete trace objects and therefore cannot disclose checking the set of visited incomplete trace objects to determine unvisited additional nets as in element j) (present in claims 2, 9, 15 and 20).

And, Roberts does not determine unvisited additional nets from a set of visited incomplete trace objects; Roberts instead determines cells based only upon information of the hierarchy database. See Roberts step 34 of FIG. 2 and col. 7, lines 40-45. Roberts does not append unvisited additional net to the list of nets, nor insert, into the set of visited incomplete trace objects, each of the unvisited additional nets; it cannot therefore anticipate requirements k) and l) (present in claims 2, 9, 15 and 20).

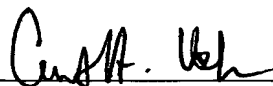
In summary, Roberts traverses cells in a hierarchical database. But Roberts does not process nets of a circuit design, and therefore cannot anticipate claims 1, 2, 8, 9, 14, 15, 19 and 20.

Reconsideration of claims 1, 2, 8, 9, 14, 15, 19 and 20 is respectfully requested.

Applicant appreciates indication of allowable subject matter in claims 3-7, 10-13, 16-18. Accordingly, reconsideration and allowance of claims 1-20 are requested.

Applicants believe no fees are due, however, should any fees be deemed necessary in connection with this Amendment, the Commissioner is hereby authorized to charge Deposit Account No. 08-2025.

Respectfully submitted,

By:   
Curtis A. Vock, Reg. No. 38,356  
LATHROP & GAGE L.C.  
4845 Pearl East Circle, Suite 302  
Boulder, CO 80301  
Telephone: (720) 931-3011  
Facsimile: (720) 931-3001